Drew Fielding SEIS 765-01 Professor Howard Due: 03/19/2025

Assignment #2

GIT Hub Repository Link: https://github.com/drewfielding4792/SEIS765-MLOPS

1. Clone the ml_ops_sentiment_lab repo:

Cloned using git clone and the url link.

```
> mlops_sentiment_lab
```

2. Remove the hard coded classes and change them to ones that are loaded from a file Removed python list of classes and created a json file called "email_class.json".

```
{} email_class.json

{]"classes": ["work", "sports", "food"
```

3. Allow users to add new classes via API. This should update the class file.

Defined a new function called load_classes()

```
def load_classes():
    with open('email_class.json', 'r') as file:
        data = json.load(file)
    return data['classes']
```

Created an endpoint called add_class to accept POST requests. It will return an error message if class already exists.

```
@app.route('/api/v1/add_class', methods=['POST'])
def add_class():
    # Get the class name from the request
    new_class = request.json.get('class_name')

# Validate that a class name is provided
if not new_class:
    return jsonify({'error': 'No class name provided'}), 400

# Load current classes
    classes = load_classes()

# Check if the class already exists
if new_class in classes:
    return jsonify({'error': f'class "{new_class}" already exists'}), 400

# Add the new class and update the classes file
    classes.append(new_class)
    update_classes(classes)

return jsonify({'message': f'class "{new_class}" added successfully', 'classes': classes}), 200
```

Here is an example of it working and appending two new classes through curl with a post method:

```
voclabs:~/environment $ curl -X POST http://localhost:3000/api/v1/add_class -H "Content-Type: application/json" -d '{"class_name": "school"}'
{
    "classes": [
    "work",
    "sports",
    "food",
    "spam",
    "finance",
    "school"
    ],
    "message": "Class \"school\" added successfully"
}
```

```
[{]"classes": ["work", "sports", "food", "spam", "finance", "school"][}]
```

```
@app.route('/api/v1/delete_class', methods=['DELETE'])
def delete_class():
    # Get the class name to delete from the request
    class_to_delete = request.json.get('class_name')

if not class_to_delete:
    return jsonify({'error': 'No class name provided'}), 400

classes = load_classes()

# Check if the class exists
if class_to_delete not in classes:
    return jsonify({'error': f'class "{class_to_delete}" not found'}), 404

# Remove the class and update the classes file
classes.remove(class_to_delete)
update_classes(classes)

return jsonify({'message': f'Class "{class_to_delete}" deleted successfully', 'classes': classes}), 200
```

I also added the capability to delete a class through Ann endpoint called delete_class to accept DELETE requests, it will provide a message of 'not found' if the class does not exist: 4. Demonstrate this working with many additional classes. Invoke classification endpoints with some different email text to show the system end 2 end.

Here are some examples of the classification working with the new classes invoked using curl:

```
voclabs:-/environment $ curl -X POST -H "Content-Type: application/json" -d '{"text": "I am a Nigerian prince. Give me $100 and I will give you $1 million dol lars in a month")' http://localhost:3000/api/v1/classify/
{
    "classifications": [
        "class": "finance",
        "similarity": 0.32852640748023987
        },
        "class": "spam",
        "similarity": 0.21189534664154053
        },
        "class": "work",
        "similarity": 0.17531776428222656
        },
        "class": "school",
        "similarity": 0.15189935266971588
        },
        "class": "food",
        "similarity": 0.12360869348049164
        },
        "class": "sports",
        "similarity": 0.09522846341133118
        },
        "message": "Email classified"
        }
        "message": "Email classified"
        }
```